```
In [22]:  from sklearn.metrics import mean_squared_error
          rmse=mean_squared_error(y_test, y_pred, squared=False)

In [23]:  rmse

Out[23]:  30.473098628830975
```

Figure.2. RMSE

2.2 SVM

For arranging and pivoting tasks, "Vector Machine Support" is a machine control strategy. Every information thing is characterized as a thing in n-layered space (where n is the quantity of things), and everything of significant worth compares to a particular area in the SVM calculation. We can now characterize them by choosing a hyperplane that obviously demonstrates the two phases (see figure underneath).
It is searching for a line or hyperplane (in an enormous space) to decrease these two courses. Prescient characterization is utilized to order whether a hyperplane is on the positive or negative side utilizing the model code displayed in Figure 3.

```
In [12]:  # print best parameter after tuning
          #print(grid.best_params_)
          re=grid.cv_results_
          #print(re)
          grid_predictions = grid.predict(X_test)

          # print classification report
          from sklearn.metrics import mean_squared_error
          rmse=mean_squared_error(y_test,grid_predictions,squared=False)

          print("The RMSE value for best parameter {}:".format(grid.best_params_),rmse)

          The RMSE value for best parameter {'C': 100, 'gamma': 'scale', 'kernel': 'rbf'}: 29.30106592611871
```

Figure.3. SVM implies early blunder

2.3 Decision tree

The testament tree is the following preparation technique that can be utilized to sort and pivot the issue. The idea of the data put away in the inside information base and the principles of affirmation are contrasted and the branches, and each leaf addresses an end [14]. The choice to get back to a tree train us to break down the idea of the item, to foresee the future like a tree, and to prove to be fruitful over the long haul. Yields/results, not set in stone by a particular assortment, however are obvious by the qualities shown in Figure 4 utilizing factual or test codes.

```
In [11]:
# print best parameter after tuning
#print(grid.best_params_)
re=grid.cv_results_
#print(re)
grid_predictions = grid.predict(X_test)

# print classification report
from sklearn.metrics import mean_squared_error
rmse=mean_squared_error(y_test,grid_predictions,squared=False)

print("The RMSE value for best parameter {}:".format(grid.best_params_),rmse)
```

```
The RMSE value for best parameter {'criterion': 'mae', 'max_features': 'auto', 'splitter': 'random'}: 40.6509638108514
```

Figure.4. RMSE endorsement tree

2.4 Random Forest

The standard vault is a framework-controlled investigating and backtracking framework. Ordinary archives can likewise be utilized to resolve in reverse issues. Since it doesn't have a line, exceptional memory is a decent decision since it can have more than one line.

2.5 Best regression model

How to Choose the Best Regression ModelRegression analysis is a dependable approach of determining whether variables have an impact on a topic of interest (in this example, our AQI values).

Which factors are most important, and which factors may be changed?omitted, and how these elements interact other can be determined with certainty regression analysis The variable in question is The dependent variable is the one that is anticipated.

It's the AQI value in our article. Factors to consider Assume that it has an impact on the dependent. The term "independent variables" refers to variables that are not reliant on one another [15].

Too few: A model that is under specified is more likely to provide biased outcomes.

Too many: An overs specified model produces estimates that are less exact.

Exactly right: The most exact estimations come from a model with the proper terms. The random forest has the most exact estimates, the least precise estimates, and no bias. As a consequence, the RANDOM FOREST is the most effective regression model.

3. Time series

Turn time is a rundown of hours gathered together consistently. The surgical table is a line. A free factor is a present moment/explicit variable that predicts the result that upholds the reason for the change. The ARMA, ARIMA, and SARIMA models assist with anticipating what's to come. Predictable data is continually changing over the long run or influences how much time it requires some investment examination.

3.1 ARMA

During the measurable grouping, the return vehicle structure (ARMA) straightforwardly clarifies the unsteady way utilizing two polynomials: one is the converse vehicle (AR), the other is the examination (MA), and by and large looks at the MA. In the model, the AR and

MA parts are equivalent (q). AR (p) is anticipated in light of early advancement. Mama (q) Figure 5 purposes a succession of clarifications and past mistakes to make presumptions. This graph shows the normal of the Autoregression movement without changing the nonstop arrangement to foresee the best heading.

```
In [17]:  result
```

Out[17]:

	Order	RMSE
0	(0, 0, 0)	49.478907
1	(0, 0, 1)	49.445758
2	(2, 0, 1)	48.844253

Figure.5. ARMA reply

3.2 ARIMA

The return vehicle shortening is a typical auto-relapse variable. A kind of factual examination utilizes time series to gather information or better get future possibilities. The factual model predicts values based on past qualities, called auto-relapse, as displayed in Figure 7. This figure shows that the coordinated moving normal chart of Autoregression has changed the arrangement into a consecutive, persistent succession.

```
In [16]:  result
```

Out[16]:

	Order	RMSE
0	(0, 0, 0)	49.478907
1	(0, 0, 1)	49.445758
2	(2, 0, 1)	48.844253
3	(1, 1, 1)	47.081949

Fig.7. ARIMA Result

3.3 SARIMA

SARIMA (ARIMA time) is the distinction between ARIMA time-explicit data gathering and time fragments, as displayed in Figure 8. Other time highlights have been added to the ARIMA model to make the ARIMA model.

```
In [16]:  result
```

Out[16]:

	Trend	Order	RMSE
0	n	(0, 0, 0)	99.900383
1	n	(0, 0, 1)	99.870882
2	n	(2, 0, 1)	44.046641
3	n	(1, 1, 1)	47.081949
4	t	(0, 0, 0)	116.668759
5	t	(0, 0, 1)	116.511437
6	t	(2, 0, 1)	56.694495
7	t	(1, 1, 1)	42.238304
8	c	(0, 0, 0)	49.478908
9	c	(0, 0, 1)	49.444351
10	c	(2, 0, 1)	49.137053
11	c	(1, 1, 1)	40.656125
12	ct	(0, 0, 0)	38.055108
13	ct	(0, 0, 1)	37.571157
14	ct	(2, 0, 1)	38.240771
15	ct	(1, 1, 1)	83.173490

Figure.8. SARIMA's response

RESULT

Quite possibly the most generally utilized expectation strategy is the ARIMA (Autoregressive Integrated Moving Average) technique: The autoregressive theory is a line of blended values and values shown in Figures 9, 10, and 11.

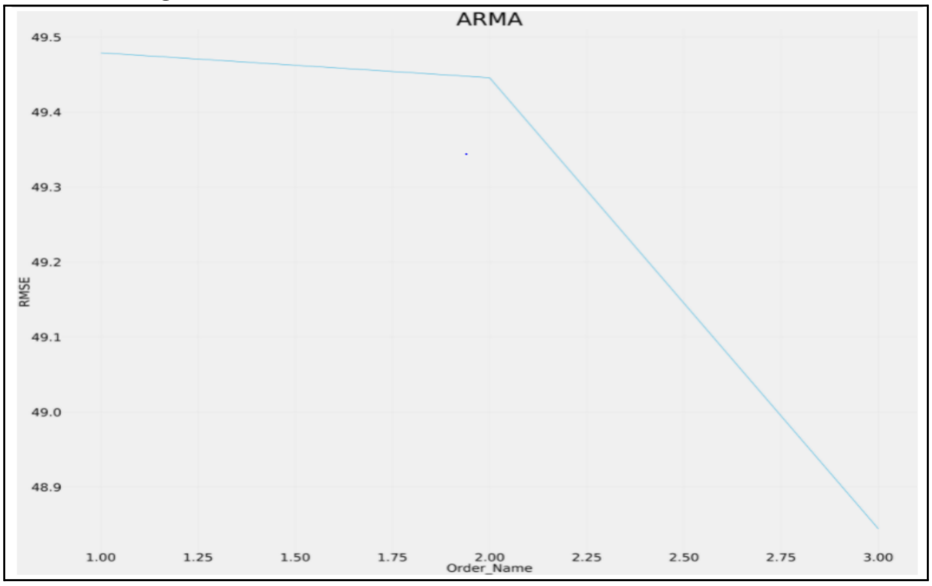

Fig.9. ARMA model

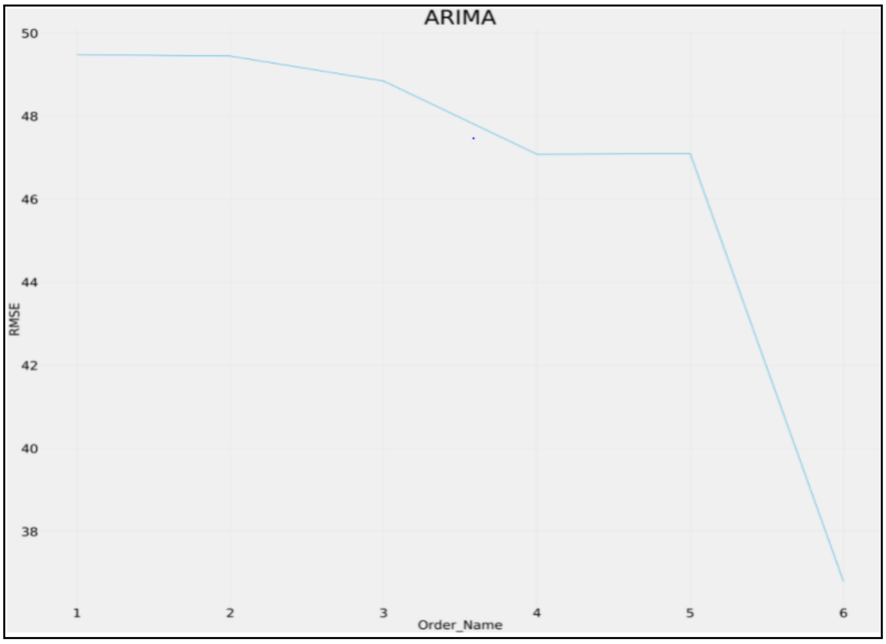

Figure 10. ARIMA plan

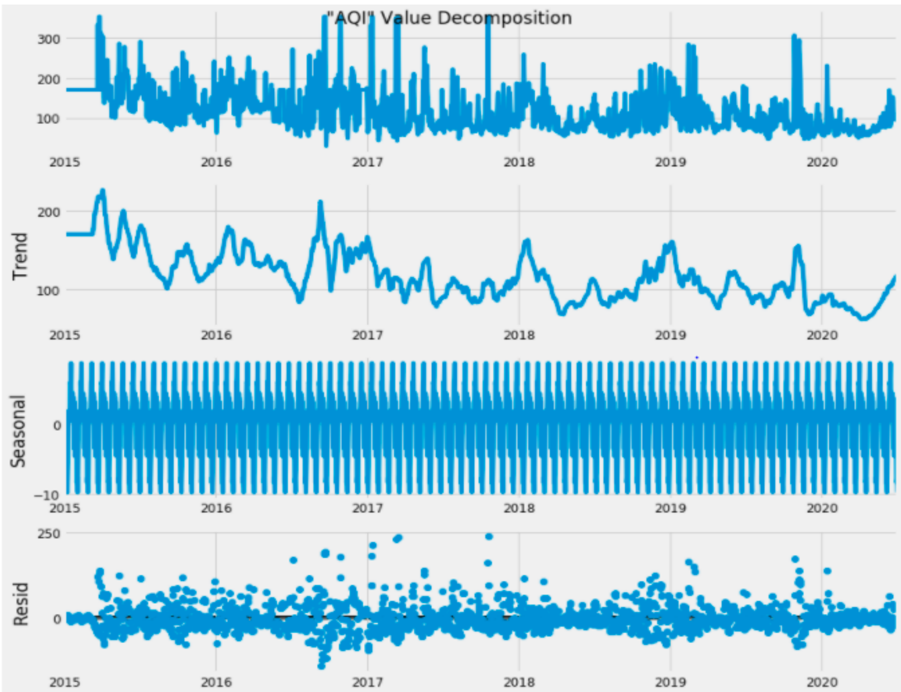

Figure.11. Time examination

4 Conclusion

In this investigation, the AQI measurements were calculated on a daily basis for three years. In addition, each pollutant's correlation coefficient was calculated. This provided a good insight into air quality and the issues we face in attempting to enhance air quality. SPM and RSPM are the principal contributors to air pollution, hence there is a pressing need to reduce their growing concentrations. Industrial activity, agricultural misconduct, and other factors might all be contributing to their rise. Although the concentrations of SO2 and NO2 appear to be under control, there has been a gradual rise in their levels, which requires ongoing monitoring.

As previously said, one of the worst contributors to human life is air pollution, which causes a variety of life-threatening disorders. Furthermore, the deterioration of the capital city reflects poorly on the country as a whole in the international arena. As a result, addressing this issue as a key priority in order to improve the level of living has become imperative.

References

1. J. K. Singh and A. K. Goel, "Prediction of Air Pollution by using Machine Learning Algorithm," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 1345-1349, doi: 10.1109/ICACCS51430.2021.9441902
2. S. Ameer et al., "Comparative Analysis of Machine Learning Techniques for Predicting Air Quality in Smart Cities," in IEEE Access, vol. 7, pp. 128325-128338, 2019, doi: 10.1109/ACCESS.2019.2925082.
3. N. Tomar, D. Patel and A. Jain, "Air Quality Index Forecasting using Auto-regression Models," 2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS), 2020, pp. 1-5, doi: 10.1109/SCEECS48394.2020.216.
4. T. Madan, S. Sagar and D. Virmani, "Air Quality Prediction using Machine Learning Algorithms –A Review," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020, pp. 140-145, doi: 10.1109/ICACCCN51052.2020.9362912..
5. J. K. Sethi and M. Mittal, "Analysis of Air Quality using Univariate and Multivariate Time Series Models," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 823-827, doi: 10.1109/Confluence47617.2020.9058303.
6. B. U. Kempaiah, B. Charan and R. J. Mampilli, "Data Analysis of Air Pollutant Levels in India," 2020 IEEE International Conference for Innovation in Technology (INOCON), 2020, pp. 1-6, doi: 10.1109/INOCON50539.2020.9298391.
7. S. S. Ganesh, S. H. Modali, S. R. Palreddy and P. Arulmozhivarman, "Forecasting air quality index using regression models: A case study on Delhi and Houston," 2017 International Conference on Trends in Electronics and Informatics (ICEI), 2017, pp. 248-254, doi: 10.1109/ICOEI.2017.8300926
8. Y. Jiao, Z. Wang and Y. Zhang, "Prediction of Air Quality Index Based on LSTM," 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2019, pp. 17-20, doi: 10.1109/ITAIC.2019.8785602.
9. Y. Yang, Z. Zheng, K. Bian, L. Song and Z. Han, "Real-Time Profiling of Fine-Grained Air Quality Index Distribution Using UAV Sensing," in IEEE Internet of Things Journal, vol. 5, no. 1, pp. 186-198, Feb. 2018, doi: 10.1109/JIOT.2017.2777820.
10. L. Ma, Y. Gao and C. Zhao, "Research on Machine Learning Prediction of Air Quality Index Based on SPSS," 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA), 2020, pp.1-5, doi: 10.1109/ICCNEA50255.2020.00011.
11. C. Srivastava, S. Singh and A. P. Singh, "Estimation of Air Pollution in Delhi Using Machine Learning Techniques," 2018 International Conference on Computing, Power

and Communication Technologies (GUCON), 2018, pp. 304-309, doi: 10.1109/GUCON.2018.8675022.
12. S. Yarragunta, M. A. Nabi, J. P and R. S, "Prediction of Air Pollutants Using Supervised Machine Learning," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1633-1640, doi: 10.1109/ICICCS51141.2021.9432078.C. Srivastava, S. Singh and A. P. Singh, "Estimation of Air Pollution in Delhi Using Machine Learning Techniques," 2018 International Conference on Computing, Power and Communication Technologies (GUCON), 2018, pp. 304-309, doi: 10.1109/GUCON.2018.8675022.
13. L. Song, "Impact Analysis of Air Pollutants on the Air Quality Index in Jinan Winter," 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, pp. 471-474, 2017.
14. V. R. Pasupuleti, Uhasri, P. Kalyan, Srikanth and H. K. Reddy, "Air Quality Prediction Of Data Log By Machine Learning," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 1395-1399, doi: 10.1109/ICACCS48705.2020.9074431.
15. S. Mahanta, T. Ramakrishnudu, R. R. Jha and N. Tailor, "Urban Air Quality Prediction Using Regression Analysis," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), 2019, pp. 1118-1123, doi: 10.1109/TENCON.2019.8929517.
16. M. Dobrea et al., "Machine Learning algorithms for air pollutants forecasting," 2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME), 2020, pp. 109-113, doi: 10.1109/SIITME50350.2020.9292238.